CLAIM AMENDMENTS

1. (Currently Amended): A method of streaming data over a network from a first device to a second device, the method comprising:

compressing the data at the first device by finding an index in a lookup table that matches an initial sequence in the data, the lookup table comprising a plurality of entries, each said entry being discoverable utilizing a particular one of a plurality of said indices, each said entry referencing whether a corresponding said index is located in a history buffer, and if so, further references one or more locations of the corresponding said index in the history buffer;

when the corresponding said entry of the matching index references a plurality of said locations:

for each said location, comparing a sequence at the location having the matching index with a sequence in the data, said sequence including the initial sequence;

deriving a matching sequence from the comparison based on at least one of a length and the location of the sequence at each said location;

representing the matching sequence using a representation that includes the length and the location of the matching sequence in the history buffer;

forming compressed data that includes at least one of said representations;

further compressing the compressed data by encoding the at least one representation that includes the length and the location of the matching

sequence, the at least one representation being encoded using a first Huffman table for encoding the length using Huffman encoding;

using a last recently used (LRU) table for encoding the location of the matching sequence in the history buffer, the LRU table listing a plurality of recently used locations of recent matching sequences;

when the location of the matching sequence is not in the LRU table, encoding the location of the matching sequence with Huffman encoding using a second Huffman table, different from the first Huffman table;

using the second Huffman table to also compress literal sequences that have no matching index in the history buffer; and

streaming the compressed data over the network to the second device.

2. (Previously Presented): A method as described in claim 1, wherein the forming compressed data includes finding one said index in the lookup table for each said sequence in the data.

3. (Original): A method as described in claim 1, wherein the corresponding said entry of the matching index references a hash chain which includes each said location of the matching index in the history buffer.

4. (Original): A method as described in claim 1, wherein the initial sequence and the index are each composed of at least two bytes.

5. (Currently Amended): A method as described in claim 1, ~~further comprising streaming the compressed data over a network,~~ wherein the data is formatted as one or more packets and the packets are compressed for ~~transmission~~ streaming over the network so that the compressing is performed on a per-packet basis.

6. (Currently Amended): A method as described in claim 1, wherein the Huffman encoding for compressing the literal sequences uses a frequency of occurrences table generated from the data and a variable length string assigned as a prefix to each literal sequence for uniquely representing the literal sequence ~~further comprising: using the second Huffman table to also compress literal sequences that have no matching index in the history buffer; and streaming the compressed literal sequences to the second device~~.

7. (Previously Presented): A method as described in claim 1, further comprising:

determining that the corresponding said entry of the matching index references a single said location;

comparing a sequence at the single said location having the matching index with the sequence in the data;

deriving a matching sequence from the comparison based on at least one of a length and the location of the sequence at the single said location; and

representing the matching sequence using a representation that includes the length and the single said location of the matching sequence in the history buffer; and

when each said sequence of the data is represented or encoded, streaming the data having the encoding or the representation.

8. (Original): A method as described in claim 1, wherein the comparison to derive the matching sequence is performed utilizing one or more thresholds selected from the group consisting of:

a number of said locations having the matching index to be compared;

a size of a value that describes each said location having the matching index; and

a size of a value that describes a length of the sequence at each said location that matches the sequence in the data that includes the matching index.

9. (Currently Amended): A method as described in claim 1, further comprising:

representing the representation as a tuple that includes a backpointer that describes the location of the matching sequence in the history buffer and the length of the matching sequence;

employing a cost function to determine if the representation utilizes less memory when stored than the matching sequence, wherein the cost function utilizes a product of a size of the backpointer and the length of the matching sequence; and ~~if so,~~

when the representation utilizes less memory when stored than the matching sequence, forming compressed data that includes the representation.

10. (Previously Presented): A method as described in claim 1, further comprising determining whether the location of the matching sequence matches one of a plurality of locations in the LRU table, wherein:

each said location in the LRU table has a corresponding said LRU representation;

each said location in the LRU table describes one of a plurality of last recently used locations of sequences in previously streamed data; and

if the location of the matching sequence is included in the LRU table, the location of the matching sequence is encoded with a corresponding said LRU representation from the LRU table.

11. (Previously Presented): One or more computer-readable storage media storing computer-executable instructions that, when executed, perform the method as recited in claim 1.

12. - 46. (Canceled)

47. (New): A method as described in claim 1, wherein:

the second device includes a client history buffer, a third Huffman table that includes codes for decoding locations of matching sequences and literal bytes, a fourth Huffman table that includes codes for decoding lengths of matching sequences, the LRU table, and a decompression module that is executable by the second device to decompress the streamed data; and

when an encoded representation is present in the configured data, the decompression module is configured to decode the representation using the LRU table, the third Huffman table, and the fourth Huffman table, and find the matching sequence in the second said history buffer based on a decoded location and a decoded length indicated by the representation.

48. (New): A computing device comprising:

one or more processors coupled to computer storage media; and

a module, stored on the computer storage media and executed on the one or more processors, to perform the following operations:

compressing data at the computing device by finding an index in a lookup table that matches an initial sequence in the data, the lookup table comprising a plurality of entries, each said entry being discoverable utilizing a particular one of a plurality of said indices, each said entry referencing whether a corresponding said index is located in a history buffer, and if so, further references one or more locations of the corresponding said index in the history buffer;

when the corresponding said entry of the matching index references a plurality of said locations:

for each said location, comparing a sequence at the location having the matching index with a sequence in the data, said sequence including the initial sequence;

deriving a matching sequence from the comparison based on at least one of a length and the location of the sequence at each said location;

representing the matching sequence using a representation that includes the length and the location of the matching sequence in the history buffer;

forming compressed data that includes at least one of said representations;

further compressing the compressed data by encoding the at least one representation that includes the length and the location of the matching sequence, the at least one representation being encoded using a first Huffman table for encoding the length using Huffman encoding;

using a last recently used (LRU) table for encoding the location of the matching sequence in the history buffer, the LRU table listing a plurality of recently used locations of recent matching sequences;

when the location of the matching sequence is not in the LRU table, encoding the location of the matching sequence with Huffman encoding using a second Huffman table, different from the first Huffman table;

using the second Huffman table to also compress literal sequences that have no matching index in the history buffer; and

streaming the compressed data over a network to another computing device.

49. (New): The computing device as described in claim 48, wherein the forming compressed data includes finding one said index in the lookup table for each said sequence in the data.

50. (New): The computing device as described in claim 48, wherein the corresponding said entry of the matching index references a hash chain which includes each said location of the matching index in the history buffer.

51. (New): The computing device as described in claim 48, wherein the initial sequence and the index are each composed of at least two bytes.

52. (New): The computing device as described in claim 48, wherein the data is formatted as one or more packets and the packets are compressed for streaming over the network so that the compressing is performed on a per-packet basis.

53. (New): The computing device as described in claim 48, wherein the Huffman encoding for compressing the literal sequences uses a frequency of occurrences table generated from the data and a variable length string assigned as a prefix to each literal sequence for uniquely representing the literal sequence.

54. (New): The computing device as described in claim 48, the operations further comprising:

determining that the corresponding said entry of the matching index references a single said location;

comparing a sequence at the single said location having the matching index with the sequence in the data;

deriving a matching sequence from the comparison based on at least one of a length and the location of the sequence at the single said location;

representing the matching sequence using a representation that includes the length and the single said location of the matching sequence in the history buffer; and

when each said sequence of the data is represented or encoded, streaming the data having the encoding or the representation.

55. (New): The computing device as described in claim 48, wherein the comparison to derive the matching sequence is performed utilizing one or more thresholds selected from the group consisting of:

a number of said locations having the matching index to be compared;

a size of a value that describes each said location having the matching index; and

a size of a value that describes a length of the sequence at each said location that matches the sequence in the data that includes the matching index.

56. (New): The computing device as described in claim 48, the operations further comprising:

representing the representation as a tuple that includes a backpointer that describes the location of the matching sequence in the history buffer and the length of the matching sequence;

employing a cost function to determine if the representation utilizes less memory when stored than the matching sequence, wherein the cost function utilizes a product of a size of the backpointer and the length of the matching sequence; and

when the representation utilizes less memory when stored than the matching sequence, forming compressed data that includes the representation.

57. (New): The computing device as described in claim 48, the operations further comprising determining whether the location of the matching sequence matches one of a plurality of locations in the LRU table, wherein:

each said location in the LRU table has a corresponding said LRU representation;

each said location in the LRU table describes one of a plurality of last recently used locations of sequences in previously streamed data; and

if the location of the matching sequence is included in the LRU table, the location of the matching sequence is encoded with a corresponding said LRU representation from the LRU table.

58. (New): The computing device as described in claim 48, wherein:

the other computing device includes a client history buffer, a third Huffman table that includes codes for decoding locations of matching sequences and literal bytes, a fourth Huffman table that includes codes for decoding lengths of matching sequences, the LRU table, and a decompression module that is executable by the other computing device to decompress the streamed data; and

when an encoded representation is present in the configured data, the decompression module is configured to decode the representation using the LRU table, the third Huffman table, and the fourth Huffman table, and find the matching sequence in the second said history buffer based on a decoded location and a decoded length indicated by the representation.

59. (New): Computer storage media having executable instructions stored thereon, the instructions, when executed by one or more processors, implementing a module to perform the following operations:

compressing data at a first computing device by finding an index in a lookup table that matches an initial sequence in the data, the lookup table comprising a plurality of entries, each said entry being discoverable utilizing a particular one of a plurality of said indices, each said entry referencing whether a corresponding said index is located in a history buffer, and if so, further references one or more locations of the corresponding said index in the history buffer;

when the corresponding said entry of the matching index references a plurality of said locations:

for each said location, comparing a sequence at the location having the matching index with a sequence in the data, said sequence including the initial sequence;

deriving a matching sequence from the comparison based on at least one of a length and the location of the sequence at each said location;

representing the matching sequence using a representation that includes the length and the location of the matching sequence in the history buffer;

forming compressed data that includes at least one of said representations;

further compressing the compressed data by encoding the at least one representation that includes the length and the location of the matching sequence, the at least one representation being encoded using a first Huffman table for encoding the length using Huffman encoding;

using a last recently used (LRU) table for encoding the location of the matching sequence in the history buffer, the LRU table listing a plurality of recently used locations of recent matching sequences;

when the location of the matching sequence is not in the LRU table, encoding the location of the matching sequence with Huffman encoding using a second Huffman table, different from the first Huffman table;

using the second Huffman table to also compress literal sequences that have no matching index in the history buffer; and

streaming the compressed data over a network to a second computing device.

60. (New): The computer storage media as described in claim 59, wherein the forming compressed data includes finding one said index in the lookup table for each said sequence in the data.

61. (New): The computer storage media as described in claim 59, wherein the corresponding said entry of the matching index references a hash chain which includes each said location of the matching index in the history buffer.

62. (New): The computer storage media as described in claim 59, wherein the initial sequence and the index are each composed of at least two bytes.

63. (New): The computer storage media as described in claim 59, wherein the data is formatted as one or more packets and the packets are compressed for streaming over the network so that the compressing is performed on a per-packet basis.

64. (New): The computer storage media as described in claim 59, wherein the Huffman encoding for compressing the literal sequences uses a frequency of occurrences table generated from the data and a variable length string assigned as a prefix to each literal sequence for uniquely representing the literal sequence.

65. (New): The computer storage media as described in claim 59, the operations further comprising:

determining that the corresponding said entry of the matching index references a single said location;

comparing a sequence at the single said location having the matching index with the sequence in the data;

deriving a matching sequence from the comparison based on at least one of a length and the location of the sequence at the single said location;

representing the matching sequence using a representation that includes the length and the single said location of the matching sequence in the history buffer; and

when each said sequence of the data is represented or encoded, streaming the data having the encoding or the representation.

66. (New): The computer storage media as described in claim 59, wherein the comparison to derive the matching sequence is performed utilizing one or more thresholds selected from the group consisting of:

a number of said locations having the matching index to be compared;

a size of a value that describes each said location having the matching index; and

a size of a value that describes a length of the sequence at each said location that matches the sequence in the data that includes the matching index.

67. (New): The computer storage media as described in claim 59, the operations further comprising:

representing the representation as a tuple that includes a backpointer that describes the location of the matching sequence in the history buffer and the length of the matching sequence;

employing a cost function to determine if the representation utilizes less memory when stored than the matching sequence, wherein the cost function utilizes a product of a size of the backpointer and the length of the matching sequence; and

when the representation utilizes less memory when stored than the matching sequence, forming compressed data that includes the representation.

68. (New): The computer storage media as described in claim 59, the operations further comprising determining whether the location of the matching sequence matches one of a plurality of locations in the LRU table, wherein:

each said location in the LRU table has a corresponding said LRU representation;

each said location in the LRU table describes one of a plurality of last recently used locations of sequences in previously streamed data; and

if the location of the matching sequence is included in the LRU table, the location of the matching sequence is encoded with a corresponding said LRU representation from the LRU table.

69. (New): The computer storage media as described in claim 59, wherein:

the second computing device includes a client history buffer, a third Huffman table that includes codes for decoding locations of matching sequences and literal bytes, a fourth Huffman table that includes codes for decoding lengths of matching sequences, the LRU table, and a decompression module that is executable by the second computing device to decompress the streamed data; and

when an encoded representation is present in the configured data, the decompression module is configured to decode the representation using the LRU table, the third Huffman table, and the fourth Huffman table, and find the matching sequence in the second said history buffer based on a decoded location and a decoded length indicated by the representation.

70. (New): A method of streaming data over a network from a first device to a second device, the method comprising:

compressing the data at the first device by finding an index in a lookup table that matches an initial sequence in the data, the lookup table comprising a plurality of entries, each said entry being discoverable utilizing a particular one of a plurality of said indices, each said entry referencing whether a corresponding said index is located in a history buffer, and if so, further references one or more locations of the corresponding said index in the history buffer;

when the corresponding said entry of the matching index references a plurality of said locations:

for each said location, comparing a sequence at the location having the matching index with a sequence in the data, said sequence including the initial sequence;

deriving a matching sequence from the comparison based on at least one of a length and the location of the sequence at each said location;

representing the matching sequence using a representation that includes the length and the location of the matching sequence in the history buffer;

forming compressed data that includes at least one of said representations, wherein forming the compressed data includes finding one said index in the lookup table for each said sequence in the data;

further compressing the compressed data by encoding the at least one representation that includes the length and the location of the matching

sequence, the at least one representation being encoded using a first Huffman table for encoding the length using Huffman encoding;

using a last recently used (LRU) table for encoding the location of the matching sequence in the history buffer, the LRU table listing a plurality of recently used locations of recent matching sequences;

when the location of the matching sequence is not in the LRU table, encoding the location of the matching sequence with Huffman encoding using a second Huffman table, different from the first Huffman table;

using the second Huffman table to also compress literal sequences that have no matching index in the history buffer, the Huffman encoding using a frequency of occurrences table generated from the data and a variable length string assigned as a prefix to each literal sequence for uniquely representing the literal sequence; and

streaming the compressed data over the network to the second device.

71. (New): The method as described in claim 70, wherein the corresponding said entry of the matching index references a hash chain which includes each said location of the matching index in the history buffer.

72. (New): The method as described in claim 70, wherein the initial sequence and the index are each composed of at least two bytes.

73. (New): The method as described in claim 70, wherein the data is formatted as one or more packets and the packets are compressed for streaming over the network so that the compressing is performed on a per-packet basis.

74. (New): The method as described in claim 70, further comprising:

determining that the corresponding said entry of the matching index references a single said location;

comparing a sequence at the single said location having the matching index with the sequence in the data;

deriving a matching sequence from the comparison based on at least one of a length and the location of the sequence at the single said location;

representing the matching sequence using a representation that includes the length and the single said location of the matching sequence in the history buffer; and

when each said sequence of the data is represented or encoded, streaming the data having the encoding or the representation.

75. (New): The method as described in claim 70, wherein the comparison to derive the matching sequence is performed utilizing one or more thresholds selected from the group consisting of:

a number of said locations having the matching index to be compared;

a size of a value that describes each said location having the matching index; and

a size of a value that describes a length of the sequence at each said location that matches the sequence in the data that includes the matching index.

76. (New): The method as described in claim 70, further comprising:

representing the representation as a tuple that includes a backpointer that describes the location of the matching sequence in the history buffer and the length of the matching sequence;

employing a cost function to determine if the representation utilizes less memory when stored than the matching sequence, wherein the cost function utilizes a product of a size of the backpointer and the length of the matching sequence; and

when the representation utilizes less memory when stored than the matching sequence, forming compressed data that includes the representation.

77. (New): The method as described in claim 70, further comprising determining whether the location of the matching sequence matches one of a plurality of locations in the LRU table, wherein:

each said location in the LRU table has a corresponding said LRU representation;

each said location in the LRU table describes one of a plurality of last recently used locations of sequences in previously streamed data; and

if the location of the matching sequence is included in the LRU table, the location of the matching sequence is encoded with a corresponding said LRU representation from the LRU table.

78. (New): The method as described in claim 70, wherein:

the second device includes a client history buffer, a third Huffman table that includes codes for decoding locations of matching sequences and literal bytes, a fourth Huffman table that includes codes for decoding lengths of matching sequences, the LRU table, and a decompression module that is executable by the second device to decompress the streamed data; and

when an encoded representation is present in the configured data, the decompression module is configured to decode the representation using the LRU table, the third Huffman table, and the fourth Huffman table, and find the matching sequence in the second said history buffer based on a decoded location and a decoded length indicated by the representation.